Introduction

Objectives

Primary Objectives With this discussion as background context, I propose the following primary objectives for this project:

Objective 1 Develop a command line program that takes in a monolithic DNN composed of n ordered layers and slices each of the layers to output a set of n models.

Objective 2 Develop a command line program that takes a set of n models and outputs a configuration of optimal subsets of layers to run on each available hardware from a list of pre configured devices in distributed systems to minimize prediction time.

Objective 3 Develop a command line program that can take in a configuration of ordered subsets of DNN layers and deploy them to a distributed set of devices according to the configuration and facilitate communication to perform prediction.

Secondary Objectives The intended target audience of this program are engineers who have a machine learning background but not necessarily a infrastructure management background. We want to minimize both (a) human input necessary to optimally deploy sliced up DNN models to a distributed system and (b) reduce the hurdles to execute the program. Therefore, I propose the following secondary objects:

Objective A Users should be agnostic about the available devices on the distributed system.

The intended target audience of this program are engineers who have a machine learning background but not necessarily an infrastructure management background. Therefore, we should not expect the user to perform any networking configuration.

Objective B Users should be agnostic about memory configuration of individual devices on the distributed system.

It is highly likely that devices within the distributed system will have varying operating system configurations and memory constraints. We want our system to detect when to-be-run subsets of layers or their inputs will not fit into memory during deployment and consequently prevent those layer subsets from being configured.

Objective C Develop a program to clean up and undo deployments.

Production ready models in the real world get fine tuned over time with more training data. A tool to bring back the deployment software to a fresh slate by undoing all actions from the previous deployment will reduce overheads for updating models.

Objective D Allow users to provide arguments during configuration to fine tune the optimization to meet the needs of the user.

Some prediction tasks require predictions on large batches of samples while some require fast prediction on small batches of samples. We want to provide the user options to fine tune the optimal deployment configuration to take factors like these into consideration.

Objective E Bundle up the command line programs into a publishable executable that will work out of the box.

This requirement is helpful because we want to minimize overheads for the user.

The program should automatically detect missing supporting services on the distributed system and spawn/despawn them accordingly even with a fresh setup of the tool.

Objective F Determine failure events after deployment and recover from those states automatically.

Distributed systems, by their nature, are prone to failure as the probability of the entire system going down increases as the number of configured devices grows. Detecting these events and recovering from these will further reduce the maintenance overhead of the software.

Accomplishments

I have developed a command line interface program called "{NAME}." This program is a collection of the following sub commands "slice" for Objective 1, "benchmark" and "configure" for Objective 2 and "deploy" for Objective 3. The commands can run in the following way